(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 617 387 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **94104706.0**

(51) Int. Cl.5: **G07F 7/08**

(22) Date of filing: **24.03.94**

(30) Priority: **24.03.93 JP 64503/93**

(43) Date of publication of application:
28.09.94 Bulletin 94/39

(84) Designated Contracting States:
DE FR GB

(71) Applicant: **Kabushiki Kaisha Toshiba**
72, Horikawa-cho
Saiwai-ku
Kawasaki-shi (JP)

(72) Inventor: **Iijima, Yasuo, c/o Intellectual Property Division**
K.K. Toshiba,
1-1 Shibaura 1-chome,
Minato-ku
Tokyo 105 (JP)

(74) Representative: **Lehn, Werner, Dipl.-Ing. et al**
Hoffmann, Eitle & Partner,
Patentanwälte,
Arabellastrasse 4
D-81925 München (DE)

(54) File management apparatus for IC card.

(57) A file management apparatus in which a memory (13) is divided into a plurality of files in an IC card (1), and the divided files are defined as upper and lower files and managed in a tree structure. The file management apparatus has an access limiting unit (11) for limiting access to one of the lower files belonging to the upper file, relaxing the access, and limiting a set of a plurality of specific files of lower files belonging to an upper file, and an access relaxing unit (11) for relaxing the limitation of the access.

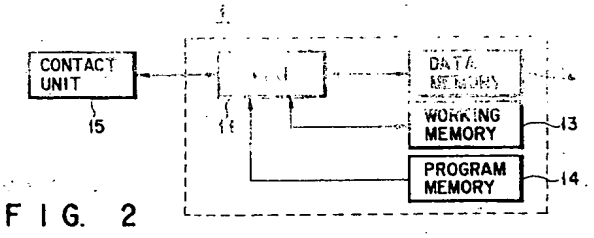

FIG. 2

EP 0 617 387 A2

The present invention relates to a file management apparatus for an IC card incorporating an IC chip having a nonvolatile memory and a control element such as a CPU for controlling the nonvolatile memory, which apparatus serves to manage a plurality of files separately set in the memory.

In recent years, as a portable data storage medium, an IC card incorporating an IC chip having a nonvolatile memory and a control element such as a CPU for controlling the memory has received a great deal of attraction.

The IC card of this type is known in U.S. Patent No. 4,985,615 having a memory whose memory area is divided in correspondence with a plurality of files. In each file, data or the like required for performing a corresponding application is stored. When an application identification name is input from a terminal device in which the IC card is inserted, the IC card can be set in a state in which only a corresponding file can be selectively used. In this manner, when a plurality of application data are divided and stored into files provided in one IC card, the IC card can be multi-purposely utilized.

In some application, a specific IC card must be inhibited from being used by the user of this IC card. For example, in a credit application, when a card user is written on a black list, the use of this card by the user must be inhibited in the credit application.

When the method described above is used, with an increase in the number of users written on the black list, the amount of operation performed to check IC cards increases. In particular, when this application is employed world-wide scale, a black list to be referred to becomes enormous, the amount of operation to check IC cards becomes conspicuously increases.

For this reason, the following method is considered. That is, in each IC card, access to a file corresponding to an application whose use must be limited is forcibly inhibited to make reference of the black list unnecessary.

It is an object of the present invention to provide a file management apparatus for an IC card, in which an enormous black list used in the execution of an application need not be referred to, and the flexibility of file management of the application is improved.

According to an aspect of the present invention, there is provided a file management apparatus in which a memory is divided into a plurality of files, and the divided files are managed such that the files are arranged to have a so-called tree structure constituted by upper and lower files, comprising: first access limiting means for limiting access to one lower file of lower files belonging to a designated upper file, second access limiting means for limiting access to a set of a plurality of specific lower files of the lower files belonging to the designated upper file, and selecting means for selecting the first access limiting means and the second access limiting means.

According to another aspect of the present invention, there can be provided a file management system in which a memory is divided into a plurality of files, and the divided files are managed such that the files are arranged to have a so-called tree structure constituted by upper and lower files, comprising first access limiting means for limiting access to one lower file of lower files belonging to a designated upper file, first access relaxing means for relaxing an access limitation performed by the firs access limiting means, second access limiting means for limiting access to a set of a plurality of specific lower files of the lower files belonging to the designated upper file, second access relaxing means for relaxing an access limitation performed by the second access limiting means, and selecting means for selecting the first access limiting means, the second access limiting means, the first access relaxing means, and the second access relaxing means.

According to the present invention, the use of one file or a plurality of files corresponding to a specific application and included in an IC card can be forcibly inhibited by instruction data externally input to the IC card. Therefore, an enormous black list need not be referred to when the application is performed. One lower file belonging to an upper file or a set of a plurality of specific lower files can be selected as an object to which use inhibition processing is performed. Therefore, the flexibility of file management for applications is improved. In addition, when the use inhibition processing is released, the corresponding application can be set to be used by the user again.

This invention can be more fully understood from the following detailed description when taken in conjunction with the accompanying drawings, in which:

FIG. 1 is a block diagram showing the arrangement of a card processing device to which an IC card according to an embodiment of the present invention is applied;

FIG. 2 is a block diagram showing an arrangement of the IC card;

FIG. 3 is a memory map showing an arrangement of a data memory;

FIG. 4 is a view showing an arrangement of a directory set in the data memory;

FIGS. 5A to 5C are views respectively showing formats of pieces of various definition information;

FIG. 6 is a flow chart for explaining an instruction data input routine;

FIGS. 7A to 7E are views respectively showing formats of various instruction data;

FIG. 8 is a flow chart for explaining a data file lock instruction routine;

FIG. 9 is a view showing a format of a data file lock value;

FIG. 10 is a view showing a format of a key area lock value;

FIG. 11 is a view showing a format of a data area lock value;

FIG. 12 is a flow chart for explaining an area lock instruction routine;

FIG. 13 is a flow chart for explaining a data area access instruction routine; and

FIG. 14 shows a relationship between file access inhibition information and area access inhibition information.

An embodiment of the present invention will be described below with reference to the accompanying drawings.

FIG. 1 is a block diagram showing an arrangement of a card processing device 10 to which an IC card serving as a portable electronic device according to this embodiment. The device 10 is used as a terminal device of a financial system, a shopping system, or the like. This card processing device 10 is constituted such that an IC card 1 can be connected to a control unit, e.g., a CPU 3 of the device 10, through a card reader/writer 2, and a keyboard 4, a CRT display device 5, a printer 6, and a floppy disk device 7 are connected to the CPU 3.

FIG. 2 shows the arrangement of an IC card 1. The IC card 1 is constituted by a CPU 11 serving as a control unit, a nonvolatile data memory 12 whose stored contents can be erased, a working memory 13, a program memory 14, and a contact unit 15 for obtaining electrical contact with the card reader/writer 2 of the device 10. Of these constituent elements, the elements (the CPU 11, the data memory 12, the working memory 13, and the program memory 14) surrounded by a dashed line in FIG. 2 are constituted by one chip and mounted in the IC card main body. The CPU 11, the data memory 12, the working memory 13, and the keyboard 4 may be constituted by a plurality of IC chips.

The data memory 12 is used to store various data, and is constituted by, e.g., an EEPROM or the like. The working memory 13 is a memory for temporarily storing processing data used when the CPU 11 performs data processing, and the working memory 13 is constituted by, e.g., a RAM or the like. The program memory 14 is constituted by, e.g., a mask ROM, and stores a program of the CPU 11 or the like.

For example, as shown in FIG. 3, the data memory 12 is divided into a directory area 121, an empty area 122, and an area group 123. The area group 123 has a plurality of data areas and a key area and can be grouped by a concept called a data file. That is, the data file is a file for simultaneously managing the data areas and the key areas used in a corresponding application.

A data area is an area for storing data such as transaction data which is read and written as needed.

These pieces of various definition information are stored in the directory 121 en bloc as shown in FIG. 5. As shown in FIG. 4, the DFSNs (file serial numbers) are automatically given to definition words, respectively, when the files are formed. In this case the files are arranged to have a so-called hierarchical structure. The CPU 11 recognizes the relationships between the files on the basis of the DFSNs and parent file serial numbers stored in the data file definition words.

For example, since a key area defined by a key area 3 definition word stored sixth (#6) has a DFSN of "01", it is understood that the key area belongs to a parent file, DF1 (data file 1).

A key area 6 definition word stored eleventh (#11) is belonging to a DF4 as described above, and this DF4 has a DFSN of "02". For this reason, it is understood that the key area 6 is belonging to a DF2.

A key area is an area used for storing, e.g., a password or the like, and is subjected to write/rewrite/collate operations. The contents of the key area cannot be read out of the IC card 1.

The control element (CPU) 11 recognizes the physical positions or the like of these files and areas by the directory 121 in the data memory 12. In order to recognize the physical positions or the like, as shown in FIGS. 5A to 5C, pieces of various definition information (to be described later) corresponding to the files and areas are stored.

FIG. 5A shows information for defining one data file. This definition information is constituted by data PTN for identifying data file definition information in the directory 121, a data file serial number DFSN assigned to this data file, a parent file serial number PFSN which is an upper file of this data file, a data file name DFN given to this data file, a name length NL representing the length of the data file name, a data file size DFS, a data file access condition DFAC representing the access condition of the data file, a data file status DFST for holding the status of the file, and bit check code data BBC for checking the validity of all the data.

The DFST shown in FIG. 5A has a data file lock value which will be described in detail later by referring to FIG. 9.

This lock value has a 2-byte format shown in FIG. 9, and different lock functions are assigned to the bits of the lock value. As shown in FIG. 9, lower two bits B1 and B2 of the first byte of the two bytes represent lock functions related to access to a data file, respectively, and subsequent two bits B3 and B4 represent lock functions related to free access to the data file, respectively.

More specifically, the meanings of the bits of the first byte are defined as follows. That is, a lowermost bit B1 designates, when an access condition is set in directory change access (e.g., a new area is additionally set in the file) to the data file, whether this access is inhibited. A second bit B2 designates, when an access condition is set in directory reference access to the data file, designates whether this access is inhibited. A third bit B3 designates, when a free condition (access can be performed regardless of a result obtained by checking whether a key is collated) is set in directory change access to the data file, whether this access is inhibited. A fourth bit B4 designates, when a free condition is set in the directory reference access to the data file, whether this access is inhibited.

Bits b7 and b5 of the first byte of the lock value represent lock functions related to change access to a key area belonging to the corresponding data file, and bits b8 and b6 represent lock functions related to reference access to the key area belonging to the data file.

The meanings of these bits are defined as follows. That is, the bit b5 designates, when an access condition is set in key change access (key setting/changing or the like) to the key area, whether this access is inhibited. The bit b6 designates, when an access condition is set in access for referring to a key state, designates whether this access is inhibited. The bit b7 designates, when a free condition is set in the key change access to the key area, whether this access is inhibited. The bit b8 designates, when a free condition is set in the access for referring to the key state, whether this access is inhibited.

Lower four bits b4 to b1 of the second byte of the lock value are related to lock functions related to access to the data area belonging to the corresponding data file.

The meanings of these bits are defined as follows. That is, the lowermost bit b1 designates, when an access condition is set in stored data change access (write/rewrite/erase access or the like for data) to the data area, whether this access is inhibited. The second bit b2 designates, when an access condition is set in stored data reference access (read access or the like for data) to data area, designates whether this access is inhibited. The third bit b3 designates, when a free condition is set in the stored data change access to the data area, whether this access is inhibited. The fourth bit b4 designates, when a free condition is set in the stored data reference access to the data area, whether this access is inhibited.

With regard to each bit, "1" means that access can be performed, and "0" means that access is inhibited.

FIG. 5B shows information for defining an area for various transaction data or the like. This definition information is constituted by data PTN for identifying area definition information in the directory 121, a data file serial number DFSN of data file to which this area belongs, an area identification number AID used when access to the area is performed, an area top address ATOP representing the top address of the area, an area size ASIZ representing an area size, an area access condition AAC representing the access condition of the area, an area status AST holding the status of the area, and the bit check code data BCC for checking the validity of all the data. The AST shown in FIG. 5B has an area lock value which will be described in detail later by referring to FIG. 11.

In each data area definition information, as shown in FIG. 11, lower 4 bits b1 to b4 have a lock value inherent in the data area in the same format as that of the lower 4 bits b1 to b4 of the second byte of the lock value in the data file definition information described above.

The relationship between the data file definition word and the key or data area definition word for defining key areas or data areas will now be described by referring to FIG. 14. As shown in FIG. 14, a key area 1 definition word shown in #2 of FIG. 4, a key area 2 definition word shown at #5, a key area 3 definition word shown at #6, a data area 1 definition word shown at #3, and a data area 2 definition word shown at #12 are defined to depend on the data file 1 definition word as shown in FIG. 14. The key area definition words are used to define the key area 1, key area 3 and key area 3, respectively, and data area definition words are used to define the data area 1 and data area 2, respectively. As can be understood from FIG. 14, the DF1 definition word includes file access inhibit information for inhibiting the access to the respective area definition words which include area access inhibit information for inhibiting the access to the respective key areas or data areas. Thus, the DF1 definition word can inhibit commonly the access to the respective areas in DF1.

FIG. 5C shows information for defining an area for stage various key data. This definition information is constituted by data PTN for identifying key area definition information in the directory 121, a serial number DFSN of a data file to which this area belongs, an identification number KID (key

identification data) used to access to the area, a KTOP (key area top address) representing the top address of the area, a KSIZ (key area size) representing an area size, a KAC (key area access condition) representing the access condition of a key, a KST (area status) holding the status of the area, and data BCC for checking the validity of all the data. The KST shown in FIG. 5C has a key area lock value which will be described in detail later by referring to FIG. 10.

In each key area definition information, as shown in FIG. 10, lower 4 bits b1 to b4 have a lock value inherent in the key area in the same format as that of the upper 4 bits b5 to b8 of the second byte of the lock value in the data file definition information of FIG. 9.

In each file, information for designating a key required to access to the corresponding file is defined. These pieces of information can be independently set in units of access types as described below.

The data files have access types related to a forcible lock operation representing inhibition of access to a file, a registering operation of an area into a file, a reference operation of directory information, and a lock releasing operation representing access release of the file.

The data areas have access types related to a reference operation of data, a write operation of data, a rewrite operation of data, and an erasing operation of data.

The key areas have access types related to a write operation of key data, a rewriting operation of key data, a lock operation of key data, and a lock operation of key data.

These pieces of access condition information specify a combination of keys present in the IC card 1 and constituted by, e.g., 4 bytes. These bytes respectively correspond to the access types, and a key having a BS (Bit for Setting the assignment of collation flag) corresponding to the position of a bit set in each byte is requested in access. Note that, when all the bits are reset, the collation states of the keys need not be checked (free access) in access corresponding to the reset bits.

A field for holding data indicating that a specific key is collated is arranged at a predetermined position of the working memory 13. In this field, a bit designated by a BS which this key has is set/reset in response to a key reference operation. Therefore, when an access command is input from the external device 10 with respect to each file or each area, the CPU 11 determines a specific one of the access types described above and extracts a byte representing an access condition corresponding to this access type. The CPU 11 checks whether a reference state requested by each bit of this byte coincides with a reference state on the working memory 13, thereby determining whether the access can be performed.

For example, when an access condition for reading data from a data area is a key A in the IC card 1, the CPU 11 checks whether this access condition is satisfied when a data read instruction for this data area is externally input to the CPU 11. If the reference status of the key A is not set, it is understood that the data area cannot be accessed.

These examinations are performed in not only read access but also another access such as write access in the same manner as described above. When instructions for a key area and a data file are input, as described above, corresponding access conditions and key reference statuses obtained at this time are confirmed.

In the IC card 1 in which the above access control is performed, inhibition of access to a data file, i.e., a lock function, will be described below.

As shown in FIG. 6, when the IC card 1 is inserted into the terminal device 10 shown in FIG. 1 to connect the contact unit 15 to the card reader/writer 2, the power supply terminal and data terminal of the contact unit 15 are connected to each other, thereby performing initialization such as a power supply operation and reset operation, i.e., electrical activation of the IC card 1. After this electrical activation is performed, the IC card 1 is set in a wait state for externally input command data. At this time, the IC card 1 continuously waits for the command data in step ST1. When the command data is input to the IC card 1, the flow advances to step ST2 to extract and interpret a function code at the top of the command data. Thereafter, the flow advances to a command routine corresponding to the interpreted result, processing is performed in the command routine, a result obtained by this processing is output, and is set in the command data wait state is set again.

In this state, if the inserted IC card 1 is subjected to use inhibition processing in this application, a data file lock command shown in FIG. 7A is input from the terminal device 10 to the IC card 1. As a result, the operation mode is changed from the operation mode shown in FIG. 6 to the data file lock command routine shown in FIG. 8.

That is, the CPU 11 identifies a data file currently set in a current state in step ST11 of FIG. 8. For this purpose, file selection command data shown in FIG. 7E and constituted by a data file selection function code, a date length LEN, and a data file name is input to the CPU 11. In this case, the CPU 11 searches the directory 121 for a data file having the same file name as the data file name input by the file selection command data. If the CPU 11 finds the data file, a corresponding DFSN is held at a predetermined position of the working memory 13. If the CPU 11 find no data file,

the information at the predetermined position is not changed. Note that, after the IC card 1 is electrically activated, this information is set to be "00".

When the data file set in a current state is found, the CPU 11, in step ST12, extracts an access condition related to the data file lock described above from access conditions corresponding to the data file, and compares the extracted access condition with the above key reference state to check whether this command is executed. If NO in step ST13, the flow advances to step ST14 to output response data representing access condition abnormality and returns to step ST1 of FIG. 6 in which the command data wait state is set.

If YES in step ST13, the CPU 11 extracts status information corresponding to the data file in step ST15, and, in step ST16, the CPU 11 calculates a logical AND between the status information and a lock value defined next to data file lock function code data designated by the command data shown in FIG. 7A.

The CPU 11 compares the lock value input by a lock instruction in step ST16 with the status of the data file set in a current state, calculates a logical AND (AND) therebetween in units of bits, and stores the obtained result at a predetermined position of data file definition information as a new data file status, i.e., a lock value, in step ST17. Note that, at this time, validity check data BBC is calculated again, and new data BCC is written. Response data indicating normal processing is output in step ST18, and the flow returns to step ST1 of FIG. 6 in which the command data wait state is set.

These lock processing operations are performed for the key area and data area belonging to the current data file.

At this time, the area lock command data shown in FIG. 7B includes ID (identification information) given to the key area and data area together with the area lock function code and the lock value. When the CPU 11 receives this command data, as shown in FIG. 12, the CPU 11 recognizes the current file in step ST21. In step 22, the CPU 11 checks by referring to the directory 121 in FIG. 3 whether the designated ID is present in an area belonging to the current data file. If the ID is not found, the flow advances from step ST23 to step ST24 to output response data representing that no designated ID is present, and returns to step ST1 in which the command data wait state is set. If YES in step ST23, the flow advances to step ST25 to refers to an access condition corresponding to lock processing designated in a designated area, thereby checking whether access can be performed. If it is determined that the access cannot be performed, the flow advances from step ST26 to step ST27 to output response data representing access condition abnormality, and returns to step ST1 in which the command data wait state i set.

If it is determined that the access can be performed, area status information corresponding to the corresponding data file is extracted in step ST28, and this area status information is compared with the lock value designated by the command data. The CPU 11 calculates a logical AND (AND) between a lock value input by a lock instruction in step ST29 and the data file set in a current state, and stores the obtained result at a predetermined position of data file definition information as a new lock value in step ST30. Note that, at this time, BBC is calculated again, and new BCC is written. In step ST31, response data indicating processing normal end is output, and the flow returns to step ST1 in which the command data wait state is set.

The relationship between an access condition and a lock value will be described below using data read access to a data area as an example with reference to the flow chart shown in FIG. 13.

When the IC card 1 is set in an command data wait state in step ST1 of FIG. 6, and access command data, shown in FIG. 7C or 7D, for read/write area data is input to the IC card 1, the CPU 11 recognizes a current file in step ST41 as in the operation shown in FIG. 8, and, in step ST42, the CPU 11 searches pieces of data area definition information which belong to the current data file for definition information having ID designated by command data. At this time, if a corresponding ID is not found, the flow advances from step ST43 to step ST44 to output response data indicating that no ID is present, and returns to the step ST1 in which the command data wait state is set.

If the ID is found, the flow advances to step ST45 to extract a lock bit representing, of access conditions set in this definition information, an access condition related to the type of the designated access.

For example, when this access is data read access, an access condition corresponding to the data read access is extracted in step ST46. When this access condition represents a free access condition, the CPU 11 refers to the fourth bit b4 of the lower 4 bits of the second byte of the lock value of the current data file in FIG. 9 and the fourth bit b4 of the lock value of the corresponding data area. In this case, when both the bits are set to be "1", the flow advances to step ST48 in which access processing is performed. When any one of the bits is "0", the flow advances to step ST47 to output response data indicating that the data area is locked, and the returns to the step ST1 in which the command wait state is set.

When the access condition requests collation of a key, the CPU 11 refers to the sixth bit b6 of

the second byte of the lock value of the current data file in FIG. 9 and the second bit b2 of the lock value of the corresponding data area in FIG. 10. When any one of the two bits is set to be "0", the CPU 11 outputs response data indicating that the area to be accessed is locked, and the flow returns to step ST1 in which the command data wait state is set. When both the bits are set to be "1", it is checked in step ST49 whether the key designated by the access condition is collated. If NO in step ST49, the CPU 11 outputs response data indicating access condition abnormality in step ST50, and the flow returns to step ST1 in which the command data wait state is set. If YES in step ST49, the flow advances to step ST51 in which read/write access processing is performed.

Note that, when the access processing is performed in step ST51, an area to be accessed is recognized by the top address and size of the data area set in the corresponding data area definition information. The CPU 11 outputs response data representing the processing result, and the flow returns to step ST1 in which the command data wait state is set.

In this manner, when data write/rewrite/erase access to a data area is to be performed, whether the access can be performed is checked using corresponding access conditions, the corresponding bits of the lock value of the data file, and the corresponding bits of the lock value of the data area to be accessed.

When access to the key area is to be performed, whether the access can be performed is checked using corresponding access conditions (set in definition information of the key area to be accessed), the corresponding bits of the lock value of the data file, and the corresponding bits of the lock value of the key area to be accessed.

When access to the data file is to be performed, whether the access can be performed is checked using access conditions (set in definition information of the data file to be accessed) corresponding to the access, the corresponding bits of the lock value of the corresponding data file.

In the embodiment described above, in checking whether access can be performed, when the access cannot be performed due to the lock value, response data representing that an object to be accessed is locked is output. However, this response data may be replaced with response data representing access condition abnormality.

In the embodiment, although the bits of a lock value can be independently set, when the bits are to be locked, bit values representing access inhibition may be sequentially set in the lowermost bit to the upper bits; when the bits are to be unlocked, bit values representing access permission may be set from the uppermost bit to the lower bits. In this case, when the set lock value is compared with the input set value to check whether a change in the set value is proper. If the change is not proper, response data representing lock value abnormality, the flow returns to step ST1 in which the command data wait state is set.

In the above embodiment, although 4-bit value is assigned as the lock value, the number of bits can be changed depending on the types of a corresponding command and the conditions of bits to be locked.

In the embodiment, command data corresponding to only an lock instruction and a corresponding flow are described. However, when a logical OR (OR) between the input lock value and the set lock value is calculated in place of the logical AND (AND) therebetween, the present invention can easily cope with command data for changing a lock value.

Although a contact unit is used to perform transmission/reception of data between an IC card and an external device, a method of performing transmission/reception of data in a noncontact state with respect to the external device by using light, an electric field, or a magnetic field may be used.

In this embodiment, although an IC card is exemplified as a memory card, the shape of the structure is not limited to a card-like shape, and the shape may be a rod- or block-like shape.

As has been described above, according to the present invention, in each IC card, the use of a data file corresponding to a specific application can be forcibly inhibited by command data externally input to the IC card. Therefore, when the application is to be performed, an enormous black list need not be referred to. One lower file belonging to an upper file or a set of a plurality of specific lower files in the hierachical structure can be selected as a target. Therefore, a file management apparatus capable of flexibly performing file management can be provided.

Claims

1. A file management apparatus in which a memory is divided into a plurality of files, and the divided files are managed in a tree structure constituted by upper and lower files belonging to the upper file, comprising:

first access limiting means for limiting access to one lower file of lower files belonging to an upper file;

second access limiting means for limiting access to a set of a plurality of specific lower files of the lower files belonging to the upper file; and

selecting means for selecting said first access limiting means and said second access

limiting means.

2. A file management apparatus in which a memory is divided into a plurality of files, and the divided files are managed in a tree structure constituted by upper and lower files belonging to the upper file, comprising:

first access limiting means for limiting access to one lower file of lower files belonging to an upper file;

first access relaxing means for relaxing an access limitation performed by said firs access limiting means;

second access limiting means for limiting access to a set of a plurality of specific lower files of the lower files belonging to the upper file;

second access relaxing means for relaxing an access limitation performed by said second access limiting means; and

selecting means for selecting said first access limiting means, said second access limiting means, said first access relaxing means, and said second access relaxing means.

3. An IC card connected to a terminal device for transmitting to and receiving from a command and data, comprising:

a memory including a plurality of files each having a plurality of areas depending to the files;

first storage means for storing access inhibit information to the files of the memory;

second storage means for storing access inhibit information to the areas depending to the files;

means for inhibiting an access operation to a specific file and the areas depending to the specific file when the access operation to the specific file is inhibited by the inhibit information stored in said first storage means; and

means for inhibiting an access operation to the areas depending to the specific file by the inhibit information stored in said second storage means when the access operation to the specific file is not inhibited by the inhibit information stored in said first storage means.

4. An IC card according to claim 3, characterized by further comprising:

first renewing means for renewing the inhibit information stored in said first storage means in accordance with a command from said terminal device; and

second renewing means for renewing the inhibit information stored in said second storage means in accordance with a command from said terminal device.

5. An IC card for receiving from and transmitting to a terminal device a command and data, comprising:

data storage means having a file and areas depending to the file for storing data to the areas;

directory information storage means for storing directory information including file definition information having position information of a file in said data storage means and access inhibit information to inhibit accessing to the file and area definition information having position information of a plurality of areas depending to the file and access inhibit information to inhibit accessing to the areas;

first determining means for determining whether the accessing to the file by referring to the inhibit information included in the file definition information;

second determining means for determining whether the accessing to the areas by referring to the inhibit information included in the area definition information;

first means for inhibiting accessing to a specific file and all the areas depending to the specific file when it is determined by said first determining means that the accessing to the specific file is inhibited;

second means for inhibiting accessing to the areas when it is determined by said first determining means that the accessing to the specific file is not inhibited and when it is determined by said second determining means that the accessing to the areas is inhibited; and

means for allowing the accessing to the areas when it is determined by said first and second determining means that the accessing is not inhibited.

6. An IC card according to claim 5, characterized in that each of said areas includes a data area for storing data and a key area for storing identification key, and the area definition information with respect to the data area and the key area is stored in the directory area.

7. An IC card according to claim 6, characterized in that the inhibit information included in the file definition information represents access inhibition to definition information of the data areas and key areas depending to the specific file, and the first means inhibits accessing to the areas when the accessing to the area definition information is inhibited by the inhibit information.

8. An IC card according to claim 7, characterized by further comprising:

first renewing means for renewing the inhibit information stored in said first storage means in response to a command from the terminal device; and

second renewing means for renewing the inhibit information stored in said second storage means in response to a command from the terminal device.

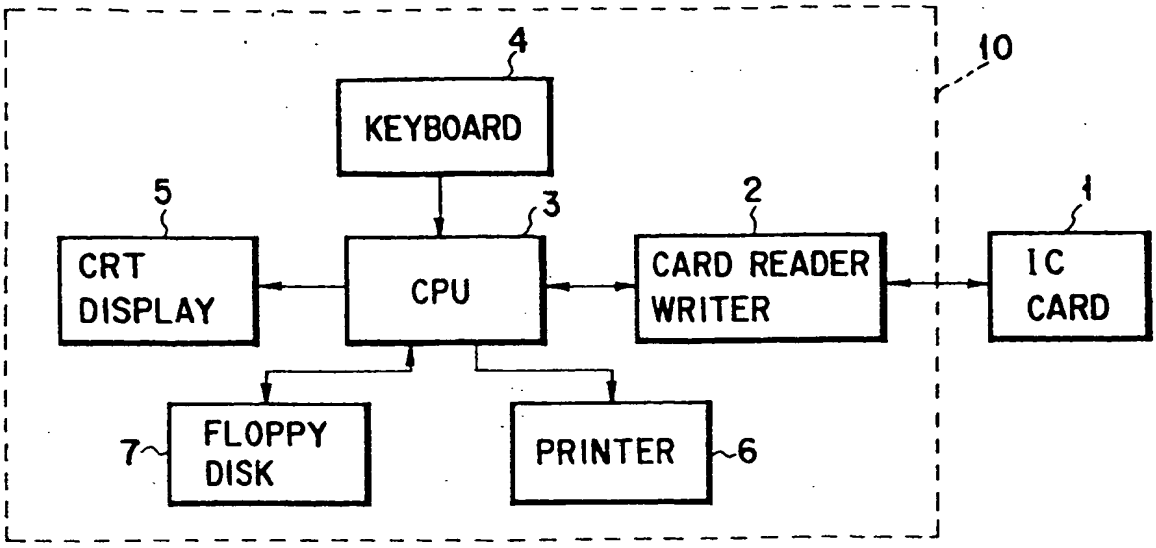

FIG. 1

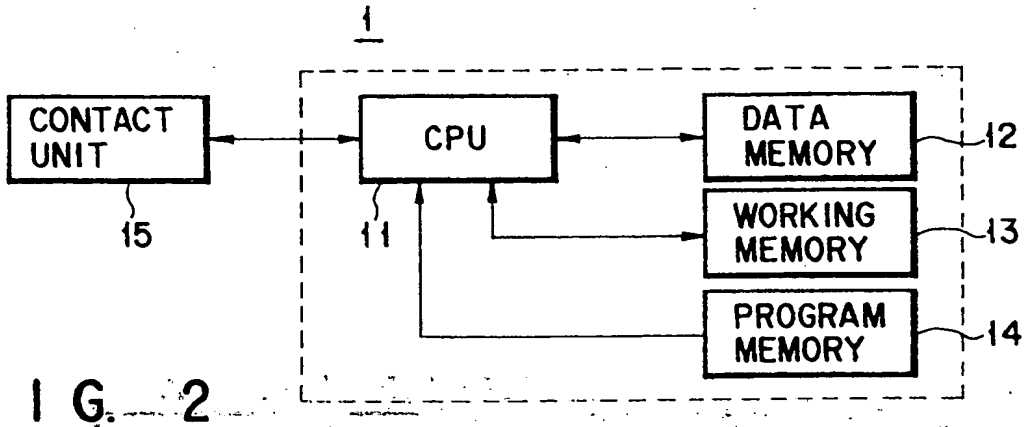

FIG. 2

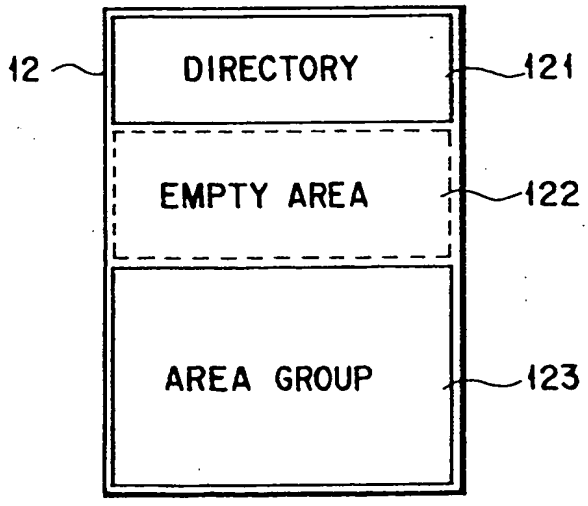

FIG. 3

PTN	DFSN	PFSN	NL	DF name	DFS	DFAC	DFST	BCC

FIG. 5A

PTN	DFSN	AID	ATOP	ASIZ	AAC	AST	BCC

FIG. 5B

PTN	DFSN	KID	KTOP	KSIZ	BS	KAC	KST	BCC

FIG. 5C

	DFSN	
#1	01	DF1 DEFINITION WORD (00)
#2	01	KEY AREA 1 DEFINITION WORD
#3	01	DATA AREA DEFINITION WORD
#4	02	DF2 DEFINITION WORD (00)
#5	02	KEY AREA 2 DEFINITION WORD
#6	01	KEY AREA 3 DEFINITION WORD
#7	00	KEY AREA 4 DEFINITION WORD
#8	03	DF3 DEFINITION WORD (02)
#9	03	KEY AREA 5 DEFINITION WORD
#10	04	DF4 DEFINITION WORD (02)
#11	04	KEY AREA 6 DEFINITION WORD
#12	01	DATA AREA DEFINITION WORD

FIG. 4

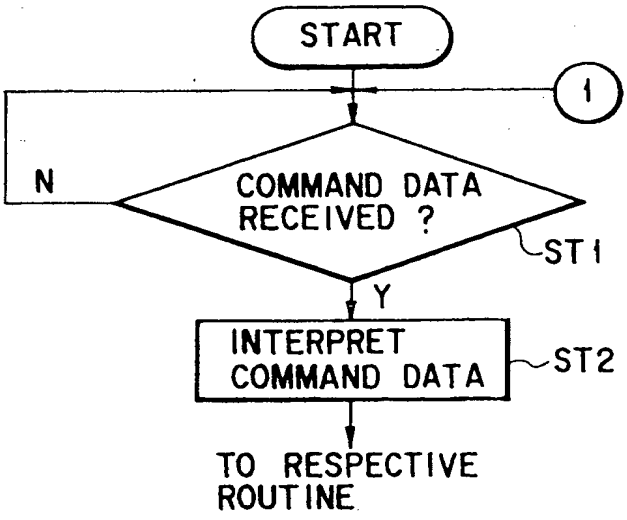

FIG. 6

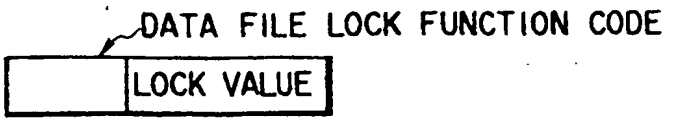

FIG. 7A

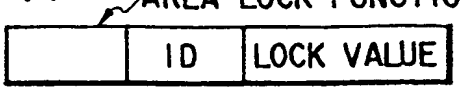

FIG. 7B

FIG. 7C

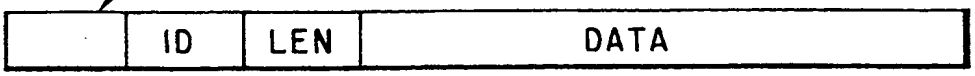

FIG. 7D

FIG. 7E

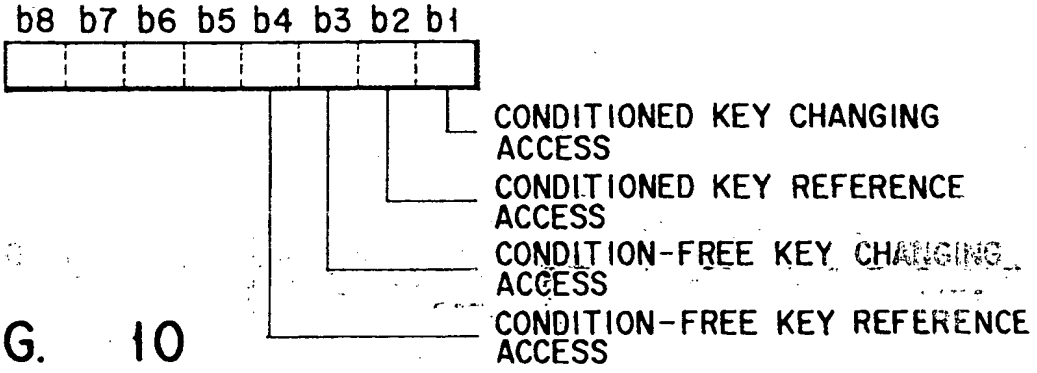

FIG. 10

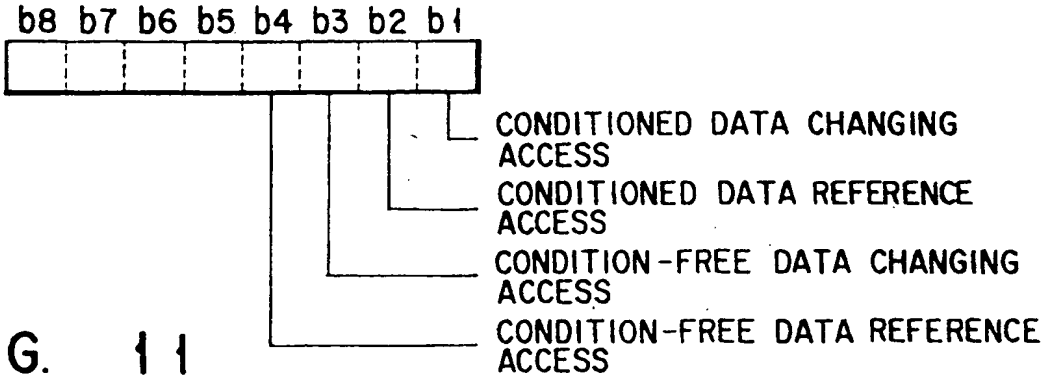

FIG. 11

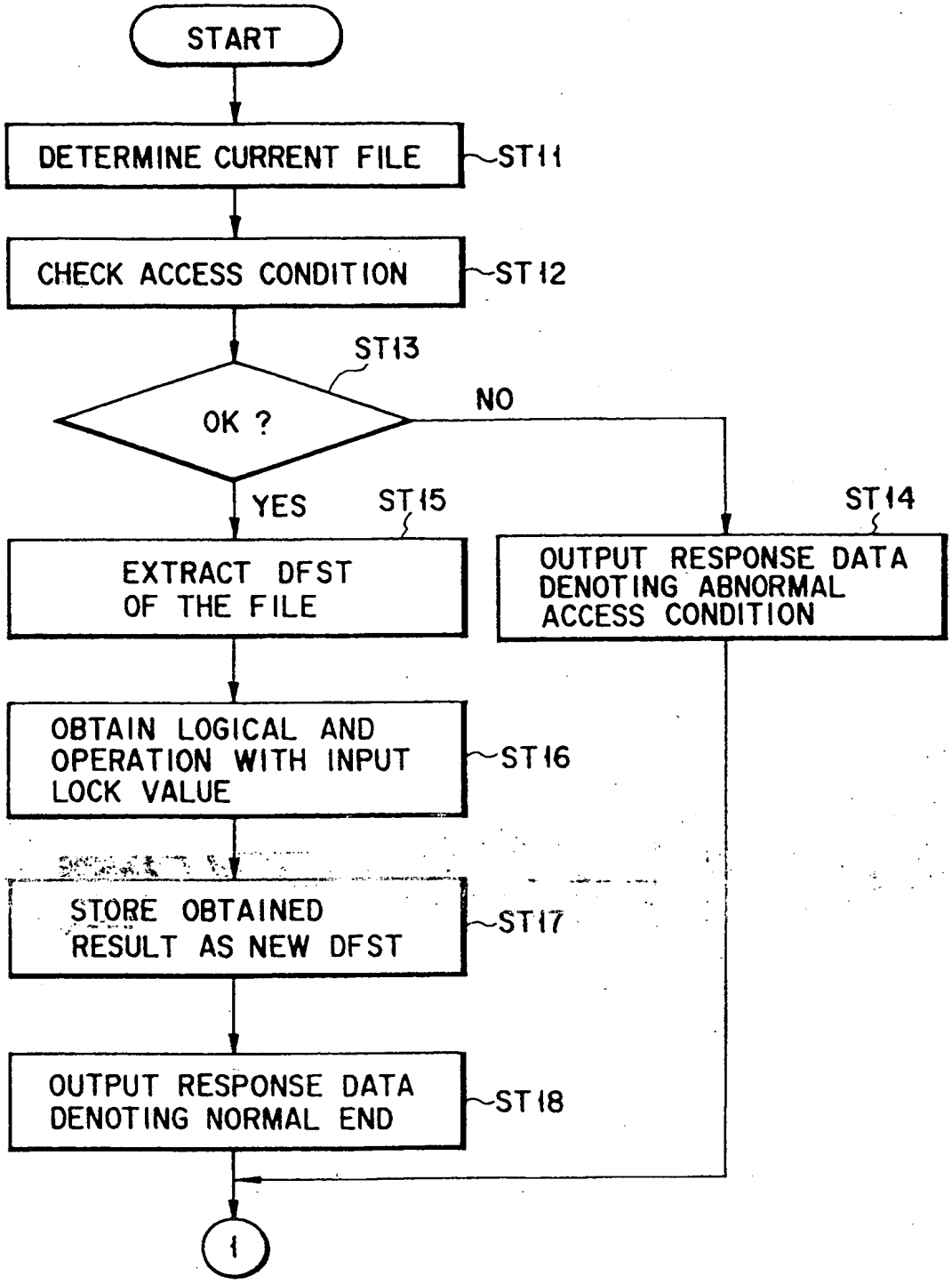

FIG. 8

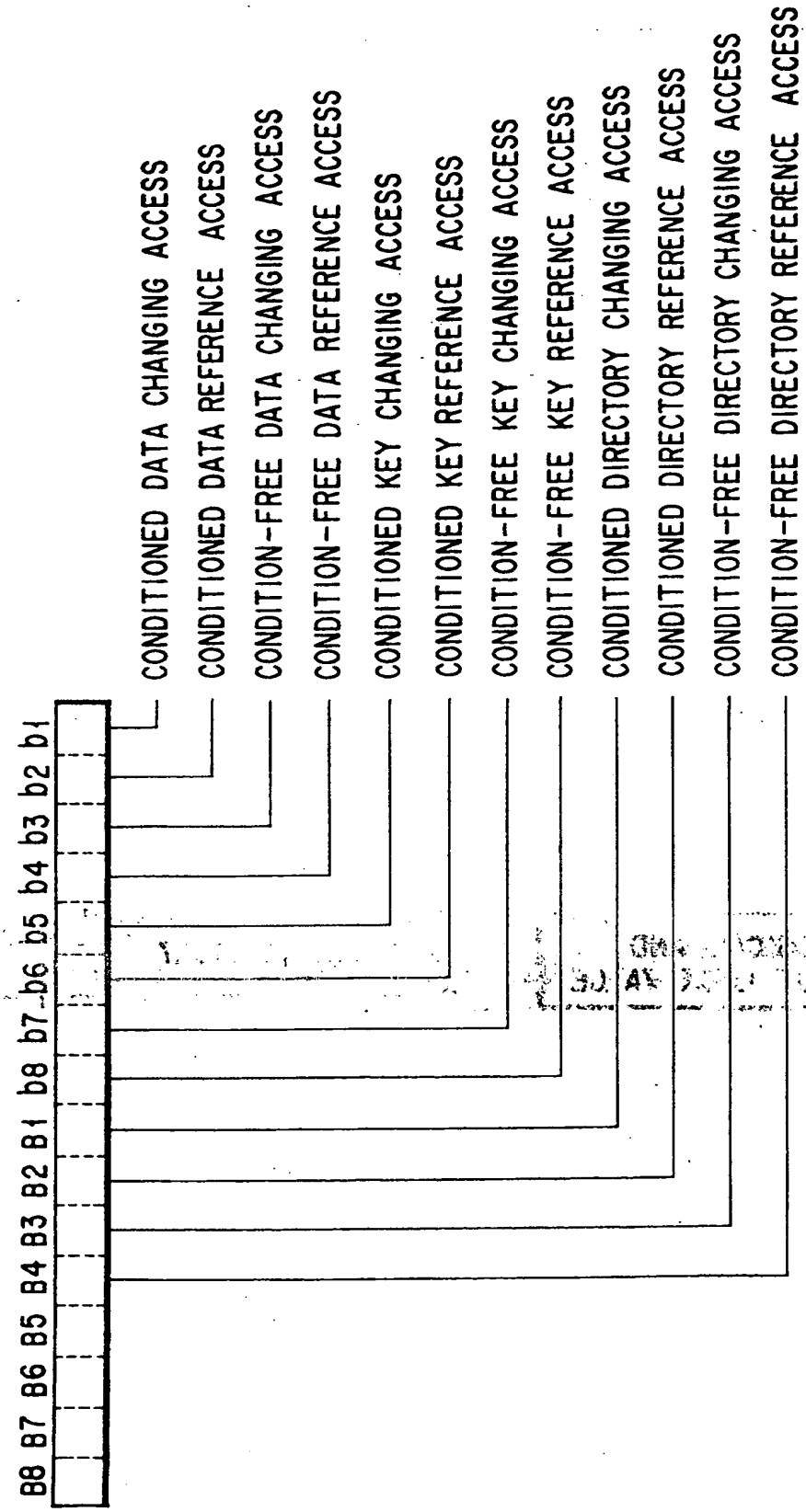

FIG. 9

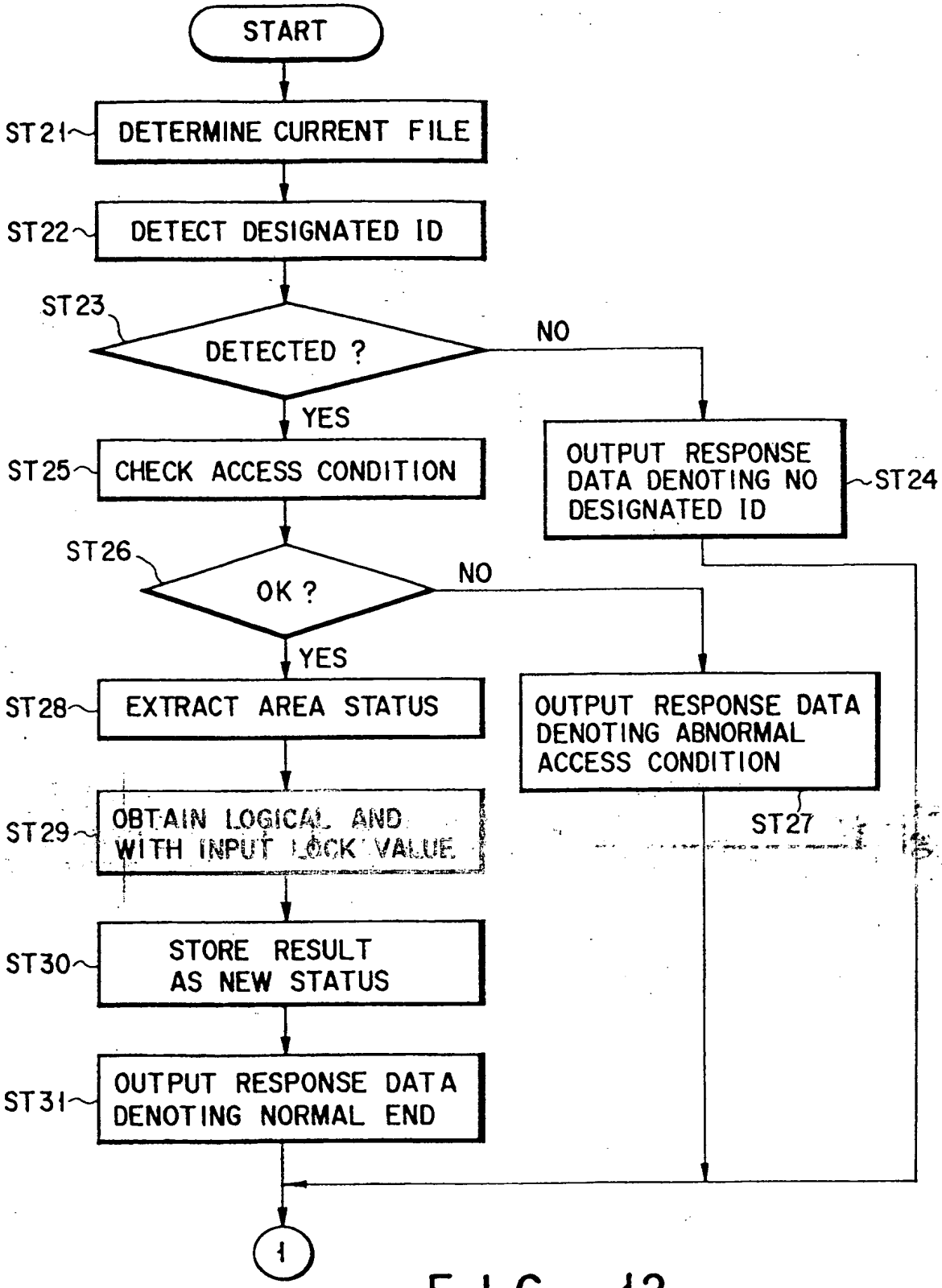

FIG. 12

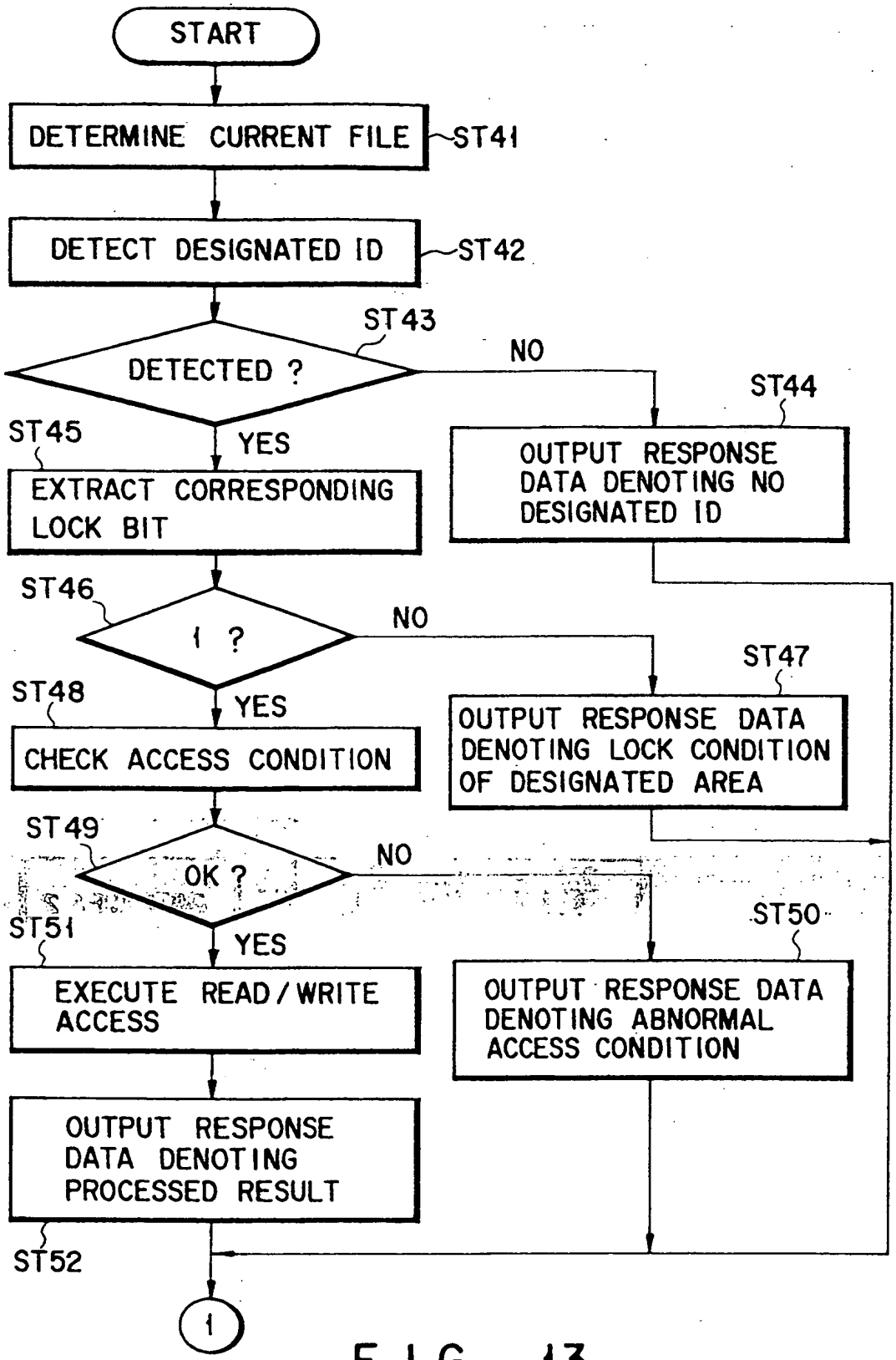

FIG. 13

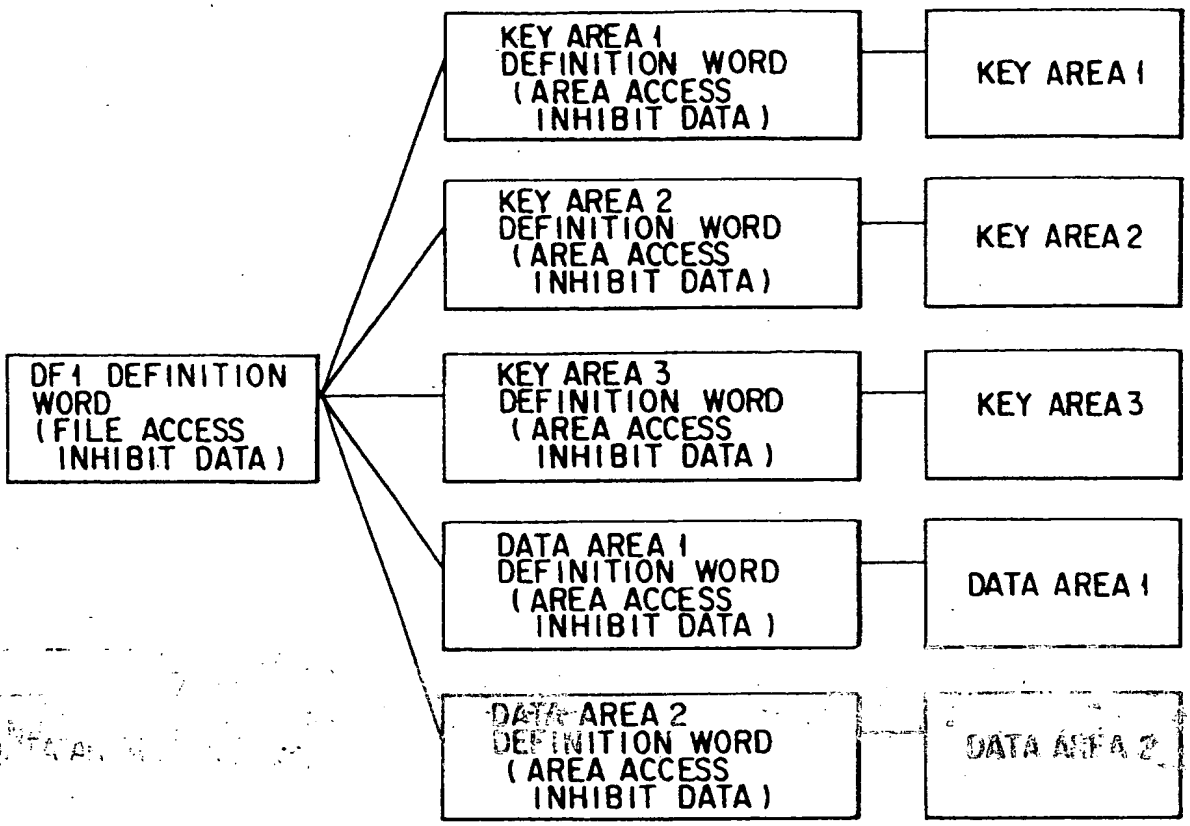

FIG. 14

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 617 387 A3**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **94104706.0**

(22) Date of filing: **24.03.94**

(51) Int. Cl.[6]: **G06K 19/073**, G07F 7/10, G07F 7/08

(30) Priority: **24.03.93 JP 64503/93**

(43) Date of publication of application:
28.09.94 Bulletin 94/39

(84) Designated Contracting States:
DE FR GB

(88) Date of deferred publication of the search report:
11.01.95 Bulletin 95/02

(71) Applicant: **KABUSHIKI KAISHA TOSHIBA**
72, Horikawa-cho
Saiwai-ku
Kawasaki-shi
Kanagawa-ken 210 (JP)

(72) Inventor: **Iijima, Yasuo, c/o Intellectual Property Division**
K.K. Toshiba,
1-1 Shibaura 1-chome,
Minato-ku
Tokyo 105 (JP)

(74) Representative: **Lehn, Werner, Dipl.-Ing. et al**
Hoffmann, Eitle & Partner,
Patentanwälte,
Arabellastrasse 4
D-81925 München (DE)

(54) File management apparatus for IC card.

(57) A file management apparatus in which a memory (12) is divided into a plurality of files in an IC card (1), and the divided files are defined as upper and lower files and managed in a tree structure. The file management apparatus has an access limiting unit (11) for limiting access to one of the lower files belonging to the upper file, relaxing the access, and limiting a set of a plurality of specific files of lower files belonging to an upper file, and an access relaxing unit (11) for relaxing the limitation of the access.

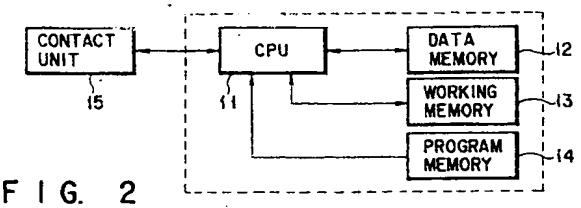

FIG. 2

EP 0 617 387 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 94 10 4706

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.5)
A	EP-A-0 262 025 (FUJITSU LTD.) * abstract; figure 4A *	1-3	G06K19/073 G07F7/10 G07F7/08
A	US-A-4 930 129 (K. TAKAHIRA) * abstract; figure 1 *	1-3,5	
A	US-A-4 829 169 (H. WATANABE) * abstract; figures 3,6-8,23 *	1-3,5	
A	US-A-4 928 001 (S. MASADA) * abstract; figure 1 *	1-3	

TECHNICAL FIELDS SEARCHED (Int.Cl.5)

G06K
G07F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	7 October 1994	Zopf, K

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)